Georgia Institute of Technology, Graphics,Visualization and Usability Center, Technical Report # GIT-GVU-98-26, August 1998.

ObjectSpaces: Context Management for Human Activity Recognition

Darnell J. Moore†, Irfan A. Essa‡, and Monson H. Hayes III†

†Center for Signal and Image Processing, School of Electrical and Computer Engineering
‡Graphics, Visualization, and Usability Center, College of Computing
Georgia Institute of Technology, Atlanta, Georgia 30332. USA.
djmoore@ece.gatech.edu, irfan@cc.gatech.edu, mhh3@ece.gatech.edu

Abstract

In this paper, we propose a vision-based method for developing computer awareness of human activities. We present an object-oriented approach called ObjectSpaces that encapsulates context into scene objects. Objects provide clues about which human motions to anticipate, making them powerful tools for discriminating actions and activities. Our hierarchical process leverages both low- and high-level representations of motion to label human interaction with objects in the surroundings. The Hidden Markov Model and Bayesian relations are used to characterize and summarize activity.

Keywords: Computer Vision, Action Recognition, Gesture Recognition, Interactive Environments.

1 Introduction

Computer vision is a critical mechanism for creating systems that can interact naturally and intelligently with people. Using computer vision techniques, we are interested in developing methodologies for recognition of human activities in an environment. Specifically, we are targeting physical interactions between a person and common articles that are apart of his or her environment. Recognition of these activities requires an understanding of motion and its relationship with objects in the scene as well as a robust means of acquiring and analyzing interactive events over time. We have developed an adaptive approach that uses context as a means of deciding the most appropriate representation for recognizing activities.

Context management plays a critical role in this process by supplying, maintaining, and discovering information about the relationships between people and objects. Objects provide clues about which human motions to anticipate, making them powerful tools for discriminating actions and activities. Our approach towards context management provides an architecture that achieves two important goals that complement each other. First, motion features extracted over time can be expressed with both a semantic and a quantifiable representation that will be useful for characterizing activities. In turn, this environmental context can be used to improve low level tracking and extraction of motion features; in effect, this approach *learns* how to track more robustly over time.

In this paper, we will present an object-oriented approach called *ObjectSpaces* to encapsulate context into scene objects. *A priori* knowledge about image contents is represented by modular, reusable objects that maintain histories of the interactivity. We also propose a method for focusing attention on motion. Low-level representations are used for tracking and detecting contact between objects and people. Once contact has been established, object context is used to guide high-level characterizations of motion for more explicit descriptions of activity. Tracking and motion analysis facilities, referred to as the *Extraction layer*, are guided by object models and parameters as well as environmental information. The *Scene layer*, which contains scene-specific context, supervises event reports from all objects. By developing a protocol for handling interactions between people and objects, we can demonstrate machine awareness of common human activities.

This work has many practical applications where passive, non-intrusive action recognition is desired, such as video surveillance and activity annotation. Moreover, work conducted in this area advances computer awareness, which is an essential step towards perceptive, intelligent interaction.

2 Related Work

As mentioned earlier, understanding the dynamics of human motion is fundamental to solving action recognition problems. For a review of motion analysis see [7]. Recently, there have been some very exciting contributions for modeling complex actions using the spatio-temporal characteristics of motion [2, 3, 8, 9, 14]. A common thread in much of the recent work in action recognition has been the use of the hidden Markov model (HMM) as a means

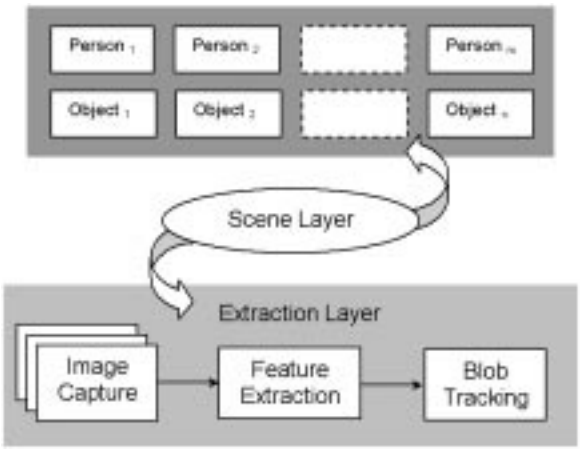

Figure 1: ObjectSpaces' interlaced multi-layer framework.

of modeling complex actions [1, 4, 5, 11, 13]. Several new frameworks that use HMMs for recognition have emerged. Specifically, Bregler evaluates motion at graduated levels of abstraction by using a 4-level decomposition framework that learns and recognizes human dynamics in video sequences [5]. While Bregler's method focuses on complex human motions, like walking, Oliver et al. present a system designed to assess interactions between people using statistical Bayesian approaches [11]. Bobick also presents several approaches to the machine perception of motion and discusses the role and levels of knowledge in each [1]. The framework proposed by Buxton et al. uses Bayesian Networks to perform surveillance in well understood scenes [6]. Our approach attempts to extend much of this work by characterizing the relationship between human motion and environmental objects.

3 Methodology

Our goal is to describe people's interactions with objects in as much detail as possible. We develop parameterized, dynamic *classes* for objects that lie in the scene using familiar object-oriented constructs. We also develop system threads, or layers, for facilitating feature extraction and scene-wide context management. In this section, we will discuss the framework for our approach, which is illustrated in Figure 1.

3.1 Defining Classes

An object-oriented process is developed because it provides a hierarchical structure that is intuitive, scalable, and reusable. Moreover, it facilitates the design and implementation of real systems. For our purposes, a class is simply a data structure, or container, for properties and methods needed for implementing class-specific tasks. All objects referenced to scene articles and people are instantiated from two parent classes, `Article` and `Person`, respectively. The object representing the scene layer is derived from a third parent class called `Scene`. The scene layer acts as manager between the extraction layer and scene articles and people.

3.1.1 The `Article` Class

We begin by establishing the n objects in the environment that we will focus on, deriving classes from `Article` for each type. Properties in this class include the centroid, bounding box, time-stamp array, state variables, and HMMs of high-level hand actions. Most of the state variables are booleans that describe features, such as "moveable" or that indicate the state, like "occluded." The array is for reporting the time and duration of contact. Actions that are indigenous to each article type are represented by HMMs and used for recognizing motion. Naturally, class-specific properties and functionality can be added to this base class.

To configure the surroundings, an initial snapshot of the background scene B is taken from the ceiling, looking downward. Using a mouse, n non-overlapping regions for each object are partitioned from B manually. Each subimage B_i, such that $B_i \subset B, 1 \leq i \leq n$, and its edge image, E_i (*from passing B_i through a Sobel edge filter*), are assigned to an object of appropriate class type. For example, Figure 4 shows separately derived classes for the keyboard, mouse, phone, etc. Using B_i and E_i, template-matching methods are adopted to detect movement or occlusion.

3.1.2 The `Person` Class

The person class works in tandem with the extraction layer to locate people based on models of a person. This view-based model is characterized by the arm/hand components as well as the head/torso component, as seen in Figure 2. The former component is characterized by physical properties, such as hand size and skin color, as well as physiological considerations, like arm span. Likewise, size and shape specify the head/torso region. Skin color is described by an array $\mathbf{C} = [\mathbf{r}\ \mathbf{g}\ \mathbf{b}]$ containing all of the flesh tones in the person's hands. This color distribution is used in the extraction layer to assist in the segmentation the hands. The `Person` class, then, is comprised of the person model along with methods for handling frame-to-frame correspondence of model components.

3.1.3 The `Scene` Class or *Scene Layer*

The scene-level layer S, derived from the `Scene` class, lies at the highest level of abstraction in the system, acting as the liaison between all other classes and the extraction layer. It maintains a list of n scene objects $\mathbf{s} = \{s_1, s_2, ...s_n\}$ and m person objects

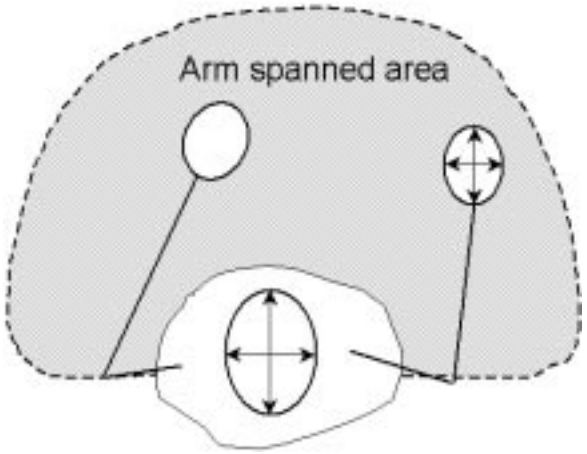

Figure 2: Structure for a person object

$\mathbf{p} = \{p_1, p_2, ...p_m\}$. We also construct an $n\mathbf{x}n$ matrix $\mathbf{K} = k_{i,j}$ where

$$k_{i,j} = \text{Prob}\{s_j(t+1)|s_i(t)\} \quad (1)$$

to hold conditional probabilities between every two articles. This layer searches for correlations between object interactions in order to classify particular activities or to identify certain human behaviors. In order to summarize contact and actions reported by objects, the scene layer applies Bayesian probabilities.

3.2 Tracking the Hands and Body

The extraction layer provides the protocol with the facilities to segment, group, label, and analyze features in each image frame. It is implemented as a primary thread in the system, running underneath the abstracted classes. `Person` and `Article` objects guide this process by supplying model parameters. Several low-level strategies that take advantage of color, motion, edge information, and model-based parameters are used for tracking and motion estimation. Higher level techniques based on HMMs are used when closer scrutiny of motion is warranted.

After classes have been defined and the surroundings configured, tracking begins by looking for the people in the scene. Image differencing between the current frame at time t and B at initialization takes place only in the portions of the image where foot traffic is permissible. The scene's motion field $F = B_t - B_0$ is used to detect and locate a person moving throughout the scene or room (using head/torso parameters). When the amount of motion appears to subside, i.e. $|B_t - B_{t-1}| < threshold$, the person's location is assumed to have stabilized. At this point, we must determine the location of a person's hands in order to deduce which items in the surroundings are handled.

Color is the basis for the recovery of the hands. We segment colored blobs from the image using the color table $\mathbf{C}$. Motion measured in the vicinity of the hands can also be used to aid in the segmentation process, but is most helpful when sweeping hand motions take place. To assist in tracking each hand, the least-squares, estimated hand position $\hat{\mathbf{h}}_{t+1}$ is given by,

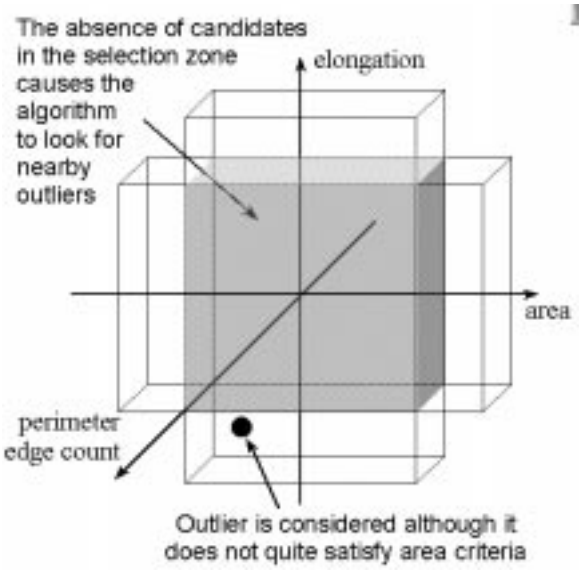

Figure 3: Softening of candidate selection

$$\hat{\mathbf{h}}_{t+1} = \begin{bmatrix} x_t + \sqrt{\frac{d^2}{(m+1)^2}} \\ y_t + m\sqrt{\frac{d^2}{(m+1)^2}} \end{bmatrix}, \quad (2)$$

where

$$m = \frac{y_t - y_{t-1}}{x_t - x_{t-1}} \text{ and } d^2 = (x_t - x_{t-1})^2 + (y_t - y_{t-1})^2.$$

This estimate is most helpful when hand movement is more pronounced; hand color shifts due to variations in the lighting conditions; or from temporary occlusions. After grouping and labeling operations, candidate blobs that do not match the profile supplied by the person model are eliminated. From the remaining candidates, blobs that appear to be nearest to previous sightings of hand objects are selected. If no candidates emerge from the regular selection process, the correspondence algorithm - with help from context supplied by the scene layer - *softens* certain model parameters to search for candidates that may be close, but slightly outside of our formal candidate selection space. For example, a blob satisfying color, position, and compactness requirements, but is too small to satisfy the initial model requirements can still be selected. This concept is illustrated in Figure 3.

3.3 Recognizing Interactions

An object is not expected to change state unless it is handled by a person. To identify any possible contact,

the image space containing the hands is compared to all of the object spaces for signs of overlap. Each class also specifies the duration the contact must last. If the contact is sufficient, the object's state changes from "dormant" to "attentive." An object remains in this state for a predetermined period of time or when contact with another object is made with the same hand.

While an object is in the attentive state, it monitors how a person's hands interact with it. If there are also actions associated with an object, hand motions must be interrogated by some characterization process. Motivated by the litany of success stories in speech and gesture recognition systems [10, 13], the hidden Markov model (HMM) was selected as the basis for action identification.

3.3.1 Hidden Markov Models

The Hidden Markov Model (HMM) can be described as a finite-state machine characterized by two stochastic processes: one process determines state transitions and is unobservable. The other produces the output observations for each state. The states are not directly determined by the observations; hence, the states are *hidden*. One of the goals of the HMM then becomes to uncover the most likely sequence of states that produced the observations. In our case, the observations are the centroids of the hands. As the hands transverse through space during some action, they pass through certain areas in the image that correspond to the states. Hand transitions from area to area, i.e. the sequence of states, are used to characterize an action.

Generally, there are three key issues associated with an HMM: training using the Baum-Welch algorithm, evaluating using the Forward-Backward algorithm, and decoding with the Viterbi algorithm. For a more complete discussion on HMMs, see [10, 12].

HMMs are well suited for recognition of complex human actions because they can efficiently characterize motion profiles in spite of the broad variation in the space and time domains in which actions are performed. If the action is identified, it is reported to the scene layer; otherwise, the object only reports that some contact took place. During the course of handling, objects that can be moved check for signs of change using stored image and edge templates. When hand contact withdraws, the object performs local template matching to determine if their was any slight movement and recalculates its new position.

Because articles detect contact, we do not have to consider all complex actions that have been modeled for a scene, only those that make the most sense. This is particularly helpful for distinguishing between families of actions that have similar motion profiles. For example, *back-and-forth* motion can be interpreted as "page flipping" by a book or "erasing" by a desk. Even though model ambiguity is minimal, model tolerances are not relaxed for fear of increasing false positives. To some extent, interactive behavior is redundant, so it is generally more favorable to omit an occurrence of action versus falsely report an event.

3.3.2 Bayesian Probability

The scene layer uses Bayesian probabilities to summarize the object activities or to find patterns between previous object contact. Human-object interactions are modeled using the Markovian assumption, e.g. *current interaction influenced by previous interaction*. While interactions with related objects naturally exhibit some level of dependency that often involves more than just the previous observation, modeling such events as a first-order Markov process preserves computational efficiency in lieu for robustness. The interaction is labeled as an action event, regardless as to whether low-level or high-level modeling (HMM) was used.

To begin, first consider a set $\mathbf{\Omega}$ of k different activities Λ, such that $\mathbf{\Omega} = \{\Lambda_1, \Lambda_2, .., \Lambda_k\}$. Each activity Λ contains a set of action models or events, i.e. $\Lambda_{writing} = \{\lambda_{drawing}, \lambda_{erasing}, \lambda_{move\ pen}\}$.

In order to compute the likelihood of an activity, we solve the relation

$$\hat{\Lambda} = \max_{\mathbf{\Omega}} \{P(\mathbf{O}|\Lambda)\}, \tag{3}$$

where $\mathbf{O}$ represents a sequence of length two of observed actions or events. The probability that a sequence of observations is produced by a given activity Λ_α is expressed as

$$\begin{aligned} P(\mathbf{O}|\Lambda_\alpha) &= \sum_{all\ \mathbf{q}} P(\mathbf{O}|\mathbf{q}, \Lambda_\alpha) P(\mathbf{q}|\Lambda_\alpha) \\ &= \sum_{all\ \mathbf{q}} P(\mathbf{O}|\mathbf{q}) P(\mathbf{q}|\Lambda_\alpha) \end{aligned} \tag{4}$$

where $\mathbf{q} = \lambda_1, \lambda_2, ..., \lambda_n$ represents an n-dimensional sequence of actions. An initial likelihood of a sequence of actions occurring during an activity $P(\mathbf{q}|\Lambda_\alpha)$ can be computed after training. Moreover, this likelihood can be updated during actual testing.

4 Experiments

To demonstrate the effectiveness of our approach, a real-time, PC-based application was developed that runs under the Win9x/NT environment. We use a ceiling-mounted color CCD camera that is pointed downward to provide a view where the location of people and objects can be clearly determined. In comparison to a lateral view of a room, this perspective is less obtrusive and less prone to occlusions caused by people moving in front of the camera (shown in Figure 4).

Figure 4: Downward view of camera with objects highlighted

4.1 Experiment I: Event Logging

We conducted a few experiments in natural environments where people interact with their surroundings. In our first experiment, we used a real office environment equipped with typical objects and appliances. We also predetermined which actions, if any, would be associated with an object in the scene.

A 6 state, *semi-ergodic* HMM with skip transitions was empirically selected to optimize recognition. Using position as features, we found the non-causal topology of this continuous HMM to be superior to strictly *left-to-right* structures. Training data consist of 10 examples for each action captured by the same person.

After configuring person and scene objects, a sequence of activities lasting over 5 minutes was choreographed and repeated 3 times. This interaction between a person and objects was observed and recorded by the system. Groundtruth observations were generated by hand and used to assess accuracy, shown in Table 1. Accuracy percentages for objects with no associated actions were based on detected contact alone. Throughout this interaction, 92% of all events were detected with 3% false positives. Because our system requires some predetermined amount of time to elapse before contact is established, some of the events were not logged. False positives were due to errors in tracking and hand correspondence. Using time-stamped logs, the system was also able to measure average time usage.

4.2 Experiment II: Behavior Analysis

In order to determine human behaviors, several smaller experiments were conducted. Because the system can discriminate between a person's left or right hand, one of the goals was to determine a right-handed subject from a left-handed subject. Nine subjects were invited to sit in front of a computer and browse the internet. The system was able to identify the correlation between the hand that interacted with the mouse and the subject's dexterous preference with 100% accuracy.

Scene Object	Time (Sec.)	Related Actions	Percent. Detected
keyboard	192	none	97%
mouse	42	none	95%
phone	55	pick up receiver,	69%
		put down receiver,	62%
		dial numbers	84%
table	72	feeding	88%
		stirring	93%
chair	301	sit down,	100%
		get up/leave	100%
bookcase	20	grab book,	88%
		return book	84%
book	26	flip forward,	89%
		flip backward	90%
desk	44	drawing,	93%
		erasing	90%
cup	12	drinking	100%

Table 1: Office objects: average elapsed time of contact, associated actions and recognition accuracy.

Figure 5: Television watching behavior

For another experiment, a scene containing a chair, TV/VCR, and 5 unlabeled VHS tapes[1] was configured to determine television watching behavior. Although physically different, the same "chair" class defined earlier was used in this scene, demonstrating the inherent ease of object reuse. However, B and E had to be redefined (see Figure 5). The tapes, each containing a

[1] *Seinfeld*:"The Finale," *The Best of Jerry Springer*, Univision's *Leonela* (soap opera in spanish), *CSPAN* coverage of the Clinton/Lewinsky scandal, and *Home Shopping Network*.

recording of a different show, were placed on different regions of the table in a particular order. Subjects were first invited to pick from among the 5 tapes until an *interesting* show was selected. Content that sustained the subject's attention for longer than 60 seconds was considered interesting. Tapes that were rejected were returned to their prior locations on the table. Using data from 18 training subjects, the system was programmed to recommend a tape to 10 test subjects by issuing recorded voice suggestions. The system's first two recommendations were accepted 90% of the time, indicating the system's ability to learn and anticipate behavior.

5 Summary and Conclusions

We have discussed a flexible and intuitive approach for exploiting environmental context to classify and recognize human activity. Contextual paradigms are helpful for designing smart vision systems because the visual information acquired can be efficiently parsed, stored, and retrieved. Object-oriented structures also facilitate the implementation of complex processes for real-time systems by offering scaleable designs. This approach can also be easily extended to multi-domain recognition of activities because of object reuse and modularity.

Although our 1-camera, view-based approach is appropriate for determining contact, several complex motions can become ambiguous. In addition, our ability to tracking fine or subtle motions was compromised due to small image resolutions (`320x240` for the entire scene). Blob tracking and correspondence was also susceptible to failures cause by shadows and other lighting conditions. The summarization procedure was also unable to identify multi-tasking scenarios when more than one activity was taking place simultaneously.

6 Future Work

In the near future, we will consider using alternative camera techniques that will allow the system to zoom into the scene to better capture complex motion. Several improvements to the extraction layer are also needed to make tracking in the presence of various lighting conditions and background clutter more robust. Heuristics and rule-based strategies will likely be added to make learning more flexible. Finally, a means of recovering multi-tasked activities is planned.

Acknowledgments

Many thanks to NCR Human Interface Technology Center, Xerox Palo Alto Reseach Center, and Ara Nefian.

References

[1] A. Bobick, "Movement, Activity, and Action: The Role of Knowledge in the Perception of Model," *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, February 1997.

[2] A. Bobick and J. Davis, "Real-time Recognition of Activity using Temporal Templates," *IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, 1996.

[3] A. Bobick and A. Wilson, "A State-Based Technique for the Summarization and Recognition of Gesture," *Proceedings of the International Conference on Computer Vision,* Cambridge, Massachusetts, 1995.

[4] M. Brand and N. Oliver, "Coupled Hidden Markov Models for Complex Action Recognition," *Proceedings of IEEE Computer Vision and Pattern Recognition*, 1997.

[5] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences," *Proceedings of IEEE Computer Vision and Pattern Recognition*, pp. 568-574, San Juan, Puerto Rico, 1997.

[6] H. Buxton and S. Gong, "Advanced Visual Surveillance using Bayesian Networks," *International Conference on Computer Vision,* Cambridge, Mass., June 1995.

[7] C. Cedras and M. Shah, "Motion-Based Recognition: A Survey," *Image and Vision Computing*, Vol. 13, No. 2, pp. 129-155, 1995.

[8] L. Davis, S. Fejes, D. Harwood, Y. Yacoob, I. Hariatoglu, and M. Black, "Visual Surveillance of Human Activity," *3rd Asian Conference on Computer Vision*, Hong Kong, China, January 1998.

[9] D. Gavrilla and L. Davis, "Tracking of Humans in Action: A 3D Model-Based Approach," *Proceedings of the IEEE Computer Vision and Pattern Recognition*, San Francisco, California, 1996.

[10] X. D. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press: Edinburgh, 1990.

[11] N. Oliver, B. Rosario, and A. Pentland, "Staistical Modeling of Human Interactions," *Proceedings from the Computer Vision and Pattern Recognition*, 1998.

[12] L. R. Rabiner, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, pp. 4-16, January, 1986.

[13] T. Starner and A. Pentland, "Visual Recognition of American Sign Language using Hidden Markov Models," *Proceedings of the International Workshop on Automatic Face and Gesture Recognition,* Zurich, Switzerland, 1995.

[14] Y. Yacoob and M. Black, "Parametertized Modeling and Recognition of Activities," *International Conf. on Computer Vision*, Mumbai-Bombay, India, January, 1998.